**Day 3 Session 2: Complete Python Variables Learning Guide**

**What is a Variable?**

A **variable** is like a labeled box that stores information you want to use later. Instead of remembering specific values, you give them names so your program can easily find and use them.

to store any number or any datsa type in python, we need variables

In Python, creating variables is straightforward - you just assign a value to a name:

age = 25

name = "Alice"

temperature = 98.6

is\_sunny = True

**What Makes Python Variables Special?**

**No Declaration Needed**

Unlike some programming languages, you don't need to announce what type of data you're storing. Python figures it out automatically.

**Dynamic Typing**

The same variable can hold different types of data:

x = 10 # x is a number

x = "hello" # now x is text

x = [1,2,3] # now x is a list

**Case Sensitive**

Name and name are completely different variables.

**Naming Rules**

* Variable names must start with a letter or underscore
* Can contain letters, numbers, and underscores
* No spaces or special characters allowed
* Cannot use Python keywords (like if, for, while, etc.)

**Understanding Variable Names**

A **variable name** is the identifier you choose to represent your data. Good variable names should be:

* Descriptive: student\_age instead of sa
* Clear: total\_price instead of tp
* Meaningful: is\_logged\_in instead of flag

**Valid Variable Names:**

age = 25

first\_name = "John"

student\_count = 30

\_private\_var = "hidden"

var2 = "second variable"

**Invalid Variable Names:**

# These will cause errors:

# 2var = 10 # cannot start with number

# first-name = "John" # cannot contain hyphens

# class = "Python" # 'class' is a reserved keyword

# my var = 5 # Cannot contain spaces

**Assignment Operators**

**Basic Assignment Operator (=)**

The **assignment operator** = is used to assign values to variables. The syntax is:

variable\_name = value

The right side (value) is evaluated first, and then assigned to the left side (variable name).

# Basic assignments

number = 42 # integer

price = 19.99 # float

message = "Hello" # string

is\_active = True # boolean

**Compound Assignment Operators**

These operators perform an operation and assignment in one step:

# Addition assignment

score = 10

score += 5 # Same as: score = score + 5

print(score) # Output: 15

# Subtraction assignment

balance = 100

balance -= 30 # Same as: balance = balance - 30

print(balance) # Output: 70

# Multiplication assignment

quantity = 3

quantity \*= 4 # Same as: quantity = quantity \* 4

print(quantity) # Output: 12

# Division assignment

total = 20

total /= 4 # Same as: total = total / 4

print(total) # Output: 5.0

# Floor division assignment

number = 17

number //= 3 # Same as: number = number // 3

print(number) # Output: 5

# Modulus assignment

remainder = 17

remainder %= 5 # Same as: remainder = remainder % 5

print(remainder) # Output: 2

# Exponentiation assignment

base = 2

base \*\*= 3 # Same as: base = base \*\* 3

print(base) # Output: 8

**Data Types on the Right Side**

When assigning values to variables, the right side can contain different data types:

**Integer (int)**

age = 25

year = 2024

negative\_num = -10

**Float (floating-point numbers)**

temperature = 98.6

price = 29.99

pi = 3.14159

**String (str)**

name = "Alice"

message = 'Hello World'

empty\_string = ""

**Boolean (bool)**

is\_sunny = True

is\_raining = False

**Lists**

numbers = [1, 2, 3, 4, 5]

names = ["John", "Jane", "Bob"]

mixed = [1, "hello", True, 3.14]

**Other Types**

# Tuple

coordinates = (10, 20)

# Dictionary

person = {"name": "Alice", "age": 30}

# Set

unique\_numbers = {1, 2, 3, 4, 5}

**Practical Examples**

**Example 1: Student Information**

# Student details

student\_name = "Sarah Johnson"

student\_id = 12345

gpa = 3.85

is\_enrolled = True

courses = ["Math", "Physics", "Chemistry"]

print(f"Student: {student\_name}")

print(f"ID: {student\_id}")

print(f"GPA: {gpa}")

print(f"Enrolled: {is\_enrolled}")

print(f"Courses: {courses}")

**Example 2: Shopping Cart**

# Shopping cart calculation

item\_price = 25.99

quantity = 3

tax\_rate = 0.08

subtotal = item\_price \* quantity

tax\_amount = subtotal \* tax\_rate

total = subtotal + tax\_amount

print(f"Item Price: ${item\_price}")

print(f"Quantity: {quantity}")

print(f"Subtotal: ${subtotal:.2f}")

print(f"Tax: ${tax\_amount:.2f}")

print(f"Total: ${total:.2f}")

**Example 3: Temperature Converter**

# Temperature conversion

celsius = 25

fahrenheit = (celsius \* 9/5) + 32

print(f"{celsius}°C is equal to {fahrenheit}°F")

# Update the temperature

celsius += 5 # Now celsius is 30

fahrenheit = (celsius \* 9/5) + 32

print(f"Updated: {celsius}°C is equal to {fahrenheit}°F")

**Example 4: User Profile**

# User profile setup

username = "coder123"

email = "user@example.com"

login\_count = 0

is\_premium = False

# User logs in

login\_count += 1

print(f"Welcome {username}! Login count: {login\_count}")

# User upgrades to premium

is\_premium = True

print(f"Premium status: {is\_premium}")

**Example 5: Game Score System**

# Game scoring

player\_name = "GameMaster"

level = 1

score = 0

lives = 3

# Player scores points

score += 100

print(f"{player\_name} - Level {level}, Score: {score}, Lives: {lives}")

# Player advances level

level += 1

score += 500

print(f"{player\_name} - Level {level}, Score: {score}, Lives: {lives}")

# Player loses a life

lives -= 1

if lives > 0:

print(f"Lives remaining: {lives}")

else:

print("Game Over!")

**Key Sticky Notes Analogy**

Think of variables as sticky notes on boxes:

* The **variable name** is the text on the sticky note
* The **assignment operator (=)** is like placing the sticky note on a box
* The **value** is what's inside the box
* You can move the sticky note to different boxes (reassign variables)

score = 100 # Sticky note "score" on box containing 100

score = 200 # Same sticky note moved to box containing 200

**Setup Instructions**

**1. Open Anaconda Prompt**

Launch the Anaconda Prompt from your system.

**2. Navigate to Your Project Directory**

Execute the following commands in sequence:

E:

cd E:\CompletePython3Bootcamp\Projects\GitDemo\PythonTutorials

**3. Activate Your Conda Environment**

conda activate PythonBootcampEnvironment

**4. Launch Jupyter Notebook**

jupyter notebook

**5. Create New Notebook**

1. Click **"New"** button in the top right corner
2. Select **"Python 3"** from the dropdown menu
3. A new untitled notebook will open

**6. Rename Your Notebook**

1. Click on **"Untitled"** at the top of the notebook
2. Rename it to Session2\_Variables
3. The file will be saved as Session2\_Variables.ipynb in the PythonTutorials directory

**Jupyter Notebook Examples from Class**

Here are the step-by-step examples we'll work through in Jupyter notebook:

**Example 1: Basic Arithmetic and Variables**

# Cell [1]: Simple addition

In [1]: 2+2

Out[1]: 4

**Example 2: Using input() Function**

# Cell [2]: Getting user input

In [2]: input("Enter first number: ")

Enter first number: 2

Out[2]: '2'

**Note:** The output '2' is in quotes, indicating it's a string, not a number.

**Example 3: Storing Input in a Variable**

# Cell [3]: Storing input in a variable

In [3]: myfirstvar = input("Enter first number: ")

Enter first number: 3

# Cell [4]: Display the variable

In [4]: myfirstvar

Out[4]: '3'

**Note:** The output '3' is still in quotes (string format).

**Example 4: Assigning Numbers Directly**

# Cell [5]: Direct number assignment

In [5]: myfirstvar = 5

# Cell [6]: Display the variable

In [6]: myfirstvar

Out[6]: 5

**Note:** The output 5 is NOT in quotes, indicating it's a number (integer).

**Example 5: Variable Operations**

# Cell [7]: Creating two variables and adding them

In [7]: mysecondvar = 6

myfirstvar+mysecondvar

Out[7]: 11

# Cell [8]: Check the data type

In [8]: type(mysecondvar)

Out[8]: int

**Example 6: Working with Decimal Numbers**

# Cell [9]: Assigning a decimal number

In [9]: mysecondvar = 6.5

myfirstvar+mysecondvar

Out[9]: 11.5

# Cell [10]: Check the new data type

In [10]: type(mysecondvar)

Out[10]: float

**Example 7: Variable Naming Rules (Errors)**

# Cell [11]: Invalid variable name with space

In [11]: my var = 5

SyntaxError: can't assign to literal

**This shows an error because variable names cannot contain spaces.**

# Cell [12]: Another invalid variable name attempt

In [12]: my var = 5

SyntaxError: invalid syntax

**Example 8: Correct Variable Naming**

# Cell [13]: Valid variable name with underscore

In [13]: myvar2 = 5

**Key Observations from the Examples:**

1. **String vs Number Output:**
   * '2' (with quotes) = string
   * 5 (without quotes) = number
2. **Data Type Changes:**
   * When mysecondvar = 6, type is int
   * When mysecondvar = 6.5, type becomes float
3. **Variable Naming Rules:**
   * my var ❌ (contains space - causes SyntaxError)
   * myvar2 ✅ (valid name with underscore)
4. **Dynamic Typing in Action:**
   * Same variable can hold different data types
   * Python automatically detects the type

**Practice Exercises**

Try these exercises in your Jupyter notebook:

**Exercise 1: Basic Variable Assignment**

# Create variables for a book

book\_title = "Python Programming"

author = "John Doe"

pages = 350

price = 29.99

in\_stock = True

# Print all information

print("Book Information:")

print(f"Title: {book\_title}")

print(f"Author: {author}")

print(f"Pages: {pages}")

print(f"Price: ${price}")

print(f"In Stock: {in\_stock}")

**Exercise 2: Using Compound Assignment Operators**

# Bank account simulation

balance = 1000.0

print(f"Initial balance: ${balance}")

# Deposit money

balance += 250.50

print(f"After deposit: ${balance}")

# Withdraw money

balance -= 75.25

print(f"After withdrawal: ${balance}")

# Apply interest (multiply by 1.02 for 2% interest)

balance \*= 1.02

print(f"After interest: ${balance:.2f}")

**Exercise 3: Variable Reassignment**

# Demonstrate dynamic typing

my\_variable = 42

print(f"Integer: {my\_variable}, Type: {type(my\_variable)}")

my\_variable = "Hello World"

print(f"String: {my\_variable}, Type: {type(my\_variable)}")

my\_variable = [1, 2, 3, 4, 5]

print(f"List: {my\_variable}, Type: {type(my\_variable)}")

my\_variable = True

print(f"Boolean: {my\_variable}, Type: {type(my\_variable)}")

**Summary**

Variables are fundamental building blocks in Python programming. They allow you to:

* Store and retrieve data efficiently
* Make your code more readable and maintainable
* Perform calculations and manipulations on data
* Create dynamic programs that can handle different inputs

Remember: Variables are your labeled storage boxes, assignment operators are how you put things in those boxes, and the values can be any type of data Python supports.

**More practices on Jupyter Notebook**

**# if we want to add two numbers without storing the numbers then we will use simple addition**

2+4

6

**# now if we want to get user input as a string even the input is a number**

input("enter any number")

'4'

**# now if we want to get user input as a number**

int(input("enter any number"))

4

**# if we want to store a number into a variable directly, it will be stored as a number only not as a string**

myvar = 6

**# the number i.e. '6' is stored into the variable 'myvar'**

**# if we want to see the number or print the number**

myvar

6

**# if we want to store a number input**

mynum1 = int(input("enter first number"))

mynum1

2

**# if we want to store another number input**

mynum2 = int(input("enter second number"))

mynum2

4

**# now simply add two numbers**

mynum1 + mynum2

6

**# now we will take only one variable**

my\_num = 2

# we will see the number

my\_num

2

**# now we will do the add operation**

**# we will add 6 with "my\_num" in different way**

**# "my\_num += 6" means "my\_num = my\_num + 6"**

my\_num += 6

# now we will print the addition output, i.e. the my\_num

my\_num

8

**# will take a variable**

myvar1 = 4

**# print the variable**

myvar1

4

**# check the type**

type(myvar1)

int

**# will take another variable**

myvar2 = 3.8

**# print the variable**

myvar2

3.8

**# check the type**

type(myvar2)

float